

Body-and-cad geometric constraint systems*

Kirk Haller
SolidWorks Corporation
300 Baker Avenue
Concord, MA 01742
khaller@solidworks.com

Audrey Lee-St. John[†]
Mount Holyoke College
South Hadley, MA 01075
astjohn@mtholyoke.edu

Meera Sitharam[‡]
University of Florida
Gainesville, FL 32611
sitharam@cise.ufl.edu

Ileana Streinu[§]
Smith College
Northampton, MA 01063
istreinu@smith.edu

Neil White
University of Florida
Gainesville, FL 32611
white@math.ufl.edu

June 8, 2010

Abstract

Motivated by constraint-based CAD software, we develop the foundation for the rigidity theory of a very general model: the *body-and-cad structure*, composed of rigid bodies in 3D constrained by pairwise coincidence, angular and distance constraints. We identify 21 relevant geometric constraints and develop the corresponding infinitesimal rigidity theory for these structures. The classical body-and-bar rigidity model can be viewed as a body-and-cad structure that uses only one constraint from this new class.

As a consequence, we identify a new, necessary, but not sufficient, counting condition for *minimal rigidity* of body-and-cad structures: *nested sparsity*. This is a slight generalization of the well-known sparsity condition of Maxwell.

*An abbreviated version appeared in: 24th Annual ACM Symposium on Applied Computing, Technical Track on Geometric Constraints and Reasoning GCR'09, Honolulu, HI, 2009. The work in this paper is based on Lee-St.John's Ph.D. dissertation [13].

[†]Partially supported by NSF grant CCF-0728783 of Ileana Streinu, an NSF Graduate Fellowship and SolidWorks Corporation.

[‡]Partially supported by a Research Grant from SolidWorks 2007.

[§]Partially supported by NSF grant CCF-0728783 and a DARPA "23 Mathematical Challenges" grant.

1 Introduction

This paper initiates the study of and sets up the foundation for the rigidity theory of a large class of 3D geometric constraint systems. These systems are composed of rigid bodies with specific **c**oincidence, **a**ngular and **d**istance constraints and are called *body-and-cad structures*. To the best of our knowledge, these constraints have not been systematically studied before from this perspective.

Motivation. Popular computer aided design (CAD) software applications based on geometric constraint solvers allow users to design complex 3D systems by placing geometric constraints among sets of rigid body building blocks. The constraints are specified by identifying *geometric elements* (points, lines, planes, or splines) on participating rigid bodies. Detecting when a user has created a fully-defined sub-system or has added a redundant (or inconsistent) constraint are important problems for providing informative feedback. However, analyzing all constraints simultaneously is a very difficult problem. In this paper, we focus on a subset of these constraints that are amenable to a rigidity-theoretical investigation.

Underlying classical rigidity theory results is a general proof pattern, spanning algebraic geometry (for *rigidity*), linear algebra (for *infinitesimal rigidity*) and graph theory (for *combinatorial rigidity*). The ultimate goal is a full combinatorial characterization of *generically minimally rigid structures*, but such results are extremely rare: 3D bar-and-joint rigidity remains a conspicuously open problem [7], while the 2D version is fully understood [12]. An important step along the way is identifying a pattern in the rigidity matrix developed as part of the *infinitesimal rigidity theory* for the structures. While this is straightforward for the well-known bar-and-joint model, it is more complicated in the body-and-bar model. In this paper, we formulate the even more involved rigidity matrix for the *body-and-cad* model.

Results. We define a *body-and-cad structure* to be composed of rigid bodies connected by *pairwise coincidence*, *angular* (parallel, perpendicular, or arbitrary fixed angular) and *distance* constraints. The constraints occur between specified points, lines or planes (called *geometric elements*). Besides the well-studied distance constraint between points (as in body-and-bar structures), we identify 20 new pairwise constraints. We label constraints by the geometric elements involved, e.g., a *line-plane* perpendicular constraint between bodies *A* and *B* indicates that a *line* on *A* is perpendicular to a *plane* on *B*. The complete set of body-and-cad constraints that we study is further subdivided into six categories:

- **Point-point constraints:** coincidence, distance.
- **Point-line constraints:** coincidence, distance.
- **Point-plane constraints:** coincidence, distance.
- **Line-line constraints:** parallel, perpendicular, fixed angular, coincidence, distance.

- **Line-plane constraints:** parallel, perpendicular, fixed angular, coincidence, distance.
- **Plane-plane constraints:** parallel, perpendicular, fixed angular, coincidence, distance.

We develop the pattern of the **rigidity matrix** and identify a necessary combinatorial counting property called *nested sparsity*, which is the counterpart of the well-known Maxwell condition [18] for fixed length rigidity. We also show that this condition is *not* sufficient. However, it can be used as a filter for finding candidate rigid components. Finally, we present an efficient algorithm for nested sparsity, based on pebble game algorithms previously developed for sparse graphs.

Related work. Classical rigidity theory [7] focuses on distance constraints between points [12] or rigid bodies [30, 34]. *Direction* constraints (where 2 points are required to define a fixed direction, with respect to a global coordinate system) are well-understood and arise from parallel redrawing applications [35]. Motivated by CAD systems, Servatius and Whiteley present a characterization, which can be viewed as a generalized Laman counting property, for 2D systems with both length and direction constraints [24].

Work on angular constraints has also focused on combinatorial characterization results. Zhou and Sitharam [37] characterize a large class of 2D angular constraint systems along with a set of combinatorial construction rules that maintain generic independence. Saliola and Whiteley [21] prove that, even in the plane, the complexity of determining the independence of a set of circle intersection angles is the same as that of generic bar-and-joint rigidity in 3D. A full characterization for angular constraints of the nature that appear in this paper is further described in [16, 13].

Combinatorial *sparsity* conditions are intimately tied with rigidity theory, appearing often as necessary conditions (as for bar-and-joint rigidity) and sometimes even as complete characterizations (as for 2D bar-and-joint and body-and-bar frameworks in arbitrary dimension) [36, 12, 30]. Pebble game algorithms have been developed for solving sparsity problems [14, 29, 15]. These algorithms do not apply, however, to the so-called $(3, 6)$ -counting conditions known to be a necessary, but not sufficient, condition for 3D bar-and-joint rigidity. In fact, no efficient algorithm is known for these counts.

Related work on the constraints studied in this paper has appeared in the CAD research community, usually within the context of decomposition approaches; a survey may be found in [25, 11]. In this setting, a geometric constraint system (GCS) is formulated as an algebraic system of equations. Due to the complexity of solving such a system, it is traditionally decomposed into structured sub-systems that can be solved and later recombined to obtain a solution to the original GCS. In the process of decomposition, approximate notions of combinatorial rigidity have been used [10, 9].

Results in the CAD literature have observed that angular constraints exhibit special behavior. For the so-called generalized Stewart platform, [5] gives explicit equations that highlight this distinction. Gao et al. [6] present a method

for analyzing 2D and 3D systems with a restricted set of coincidence, angular and distance constraints. Both [5] and [6] treat angular constraints separately, implicitly using natural necessary counting conditions to do so. We consider analogous systems from the rigidity theory perspective, expressing them infinitesimally using Grassmann-Cayley algebra; the shape of the rigidity matrix described in Section 3 explicitly reveals the distinct treatment of angular constraints. Grassmann-Cayley, Clifford algebras and geometric algebras often appears in the context of CAD or geometric theorem proving; see, e.g., [17, 23]. Recent work of [20] expresses constraints from a similar perspective when providing a foundation for software to build a GCS.

Incidence constraints have been studied previously in connection with Geometric Theorem Proving [17, 19] for projective incidence theorems. Sitharam et al. [26, 27, 28] formalize the question of obtaining a well-formed and optimal system of algebraic equations to resolve a collection of incident rigid bodies. [26] studies “well-formedness,” a condition necessary to avoid dependent equations, and a new, underlying matroid whose independent sets capture this. A combinatorial measure of algebraic complexity of the system of equations is described in [27], and another underlying matroid is used to optimize this measure. In [28], it is shown how to reconcile the independent sets of the previous two matroids to obtain an optimal, well-formed system.

Structure. Section 2 gives a brief overview of the required mathematical background. Section 3 develops the foundations for the infinitesimal rigidity theory, providing the basic building blocks used for each new constraint. Each of the full set of constraints is then expressed using these building blocks in Section 4, resulting in the complete derivation of the rigidity matrix. Section 5 identifies a new combinatorial property resulting from the structure of the rigidity matrix; this *nested sparsity* condition, while necessary, is shown not to be sufficient with a counterexample. Section 6 presents algorithms for nested sparsity using pebble games as oracles. Finally, Section 7 discusses extensions, applications and future directions.

2 Preliminaries

Our results rely on the same mathematical background as the work on body-and-bar rigidity by Tay [30] and White and Whiteley [34]. We use Grassmann-Cayley algebra, Plücker coordinates and instantaneous screw theory (see, e.g., [32, 33] and [22]). For self-containment, we briefly introduce notation and basic concepts from the Grassmann-Cayley algebra and its correspondence with instantaneous screws.

2.1 Terminology and notation.

We restrict ourselves to dimension 3 in this paper; *2-tensors* in the Grassmann-Cayley algebra (see, e.g., [32, 33]) are identified with vectors in \mathbb{R}^6 . The Grassmann-Cayley *join* operator is represented with \vee . The join $\mathbf{p} \vee \mathbf{q}$ of two

vectors $\mathbf{p}, \mathbf{q} \in \mathbb{R}^4$ is the collection of all 6 minors of the matrix M obtained with \mathbf{p} and \mathbf{q} as its rows. We fix a convention at this point to order the minors in a 6-vector as $(|M_{14}|, |M_{24}|, |M_{34}|, |M_{23}|, -|M_{13}|, |M_{12}|)^1$. The dot product of two vectors u and v is denoted $\langle u, v \rangle$. The *star operator* $*$ swaps the first and last 3 coordinates of a 6-vector. If $\mathbf{p} \in \mathbb{R}^3$ and $c \in \mathbb{R}$, we denote by $(\mathbf{p} : c)$ the vector of length 4 obtained by appending c to \mathbf{p} .

Rigid body motions. The theory of screws was introduced by Ball [1] as a way of expressing rigid body motion. Rigid body transformations are associated with elements of the special Euclidean group $\text{SE}(3)$. By Chasles' Theorem from 1830 (see [22]), they can also be expressed as *screw motions* (see Figure 1a). It follows that every instantaneous rigid body motion can be expressed as an *instantaneous screw motion* (see Figure 1b); for further details, we refer the reader to a standard text, e.g., page 24 of [22]. Both screw motions and

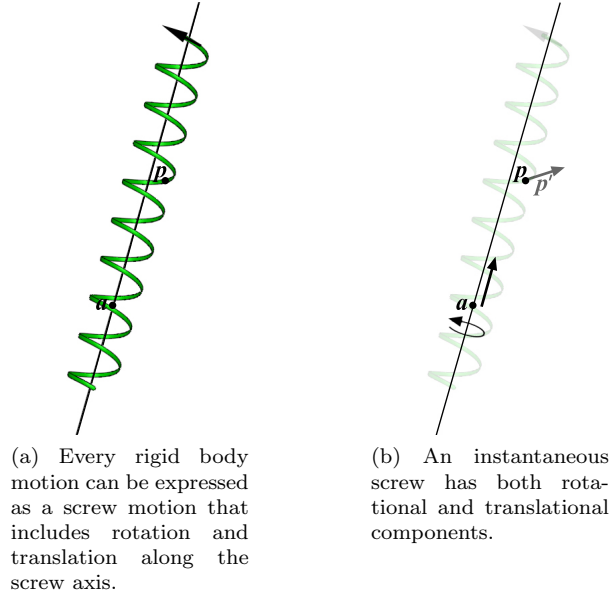


Figure 1: A screw motion and its associated instantaneous screw motion.

instantaneous screw motions are defined with respect to a *screw axis* along with a rotation about the axis and a translation along it.

In this paper, we are concerned only with instantaneous screw motions, which, for brevity, will be referred to as *instantaneous screws*. An instantaneous screw is represented by a 6-vector $\mathbf{s} = (-\boldsymbol{\omega}, \mathbf{v})$, where $\boldsymbol{\omega}, \mathbf{v} \in \mathbb{R}^3$; the minus sign in front of $\boldsymbol{\omega}$ is a convenient, technical convention. The first component $\boldsymbol{\omega}$ encodes the angular velocity; as a vector, $\boldsymbol{\omega}$ gives the direction of the screw axis,

¹We remark that other papers (e.g., [30] and [34]), use a different convention by fixing the order as $(|M_{12}|, |M_{13}|, |M_{14}|, |M_{23}|, |M_{24}|, |M_{34}|)$

and its magnitude encodes the angular speed. The translational velocity can be computed from $\boldsymbol{\omega}$ and \mathbf{v} , but we skip the details as they are not relevant for the rest of the paper. Note that the star operator applied to a screw $\mathbf{s} = (-\boldsymbol{\omega}, \mathbf{v})$ gives $\mathbf{s}^* = (\mathbf{v}, -\boldsymbol{\omega})$. There is an exact correspondence between 2-tensors and instantaneous screws. This correspondence is the key to describing the rigidity matrix of the body-and-cad structures.

2.2 Body-and-cad structures

A body-and-cad structure in 3D is composed of n bodies interconnected by pairwise constraints. Each body i is represented by a frame of reference, specified by a transformation matrix T_i from the special Euclidean group $SE(3)$. Each body i additionally has a set of *geometric elements* (points, lines or planes) identified as attachments for the constraints.

Representation of geometric elements. Each geometric element is rigidly affixed to a body i and is described with coordinates that are local with respect to the frame of reference for body i . For ease of analysis, we represent a plane in point-normal form as the pair (\mathbf{p}, \mathbf{d}) , with $\mathbf{p}, \mathbf{d} \in \mathbb{R}^3$, where \mathbf{p} is a point on the plane and \mathbf{d} is the normal to the plane. We represent a line in parametric form, given by the pair (\mathbf{p}, \mathbf{d}) , with $\mathbf{p}, \mathbf{d} \in \mathbb{R}^3$, where \mathbf{p} is a point on the line and \mathbf{d} is its direction.

2.2.1 Cad graphs

We now introduce the *cad graph*, our main combinatorial object for body-and-cad rigidity.

To illustrate this concept, consider the following example, depicted in Figure 2. Let A and B be two dice rigidly stacked with the following constraints: (i) (**plane-plane parallel**) A 's Face 1 is parallel to B 's Face 1, (ii) (**plane-plane perpendicular**) A 's Face 2 is perpendicular to B 's Face 3, (iii) (**line-plane distance**) The distance between A 's Line 12 (intersection of Faces 1 and 2) and B 's Face 1 is 1, and (iv) (**point-point coincidence**) A 's Corner 236 (the point defined by Faces 2, 3 and 6) is coincident to B 's Corner 123. These constraints are captured by a graph with two nodes connected by annotated edges called the *cad graph*; see Figure 3.

Formally, a *cad graph* (G, c) is a multigraph $G = (V, E)$ together with an edge coloring function $c : E \rightarrow C$, where $C = \{c_1, c_2, \dots, c_{21}\}$ consists of 21 colors corresponding to the full set of cad constraints:

1. point-point coincidence
2. point-point distance
3. point-line coincidence
4. point-line distance
5. point-plane coincidence

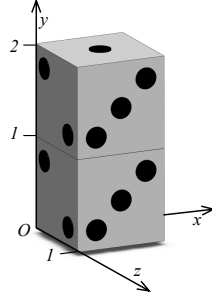


Figure 2: Two dice rigidly stacked; die A is above B . Faces are labeled by the number of dots, and face 6 lies at the bottom (opposite 1). The length of an edge is 1.

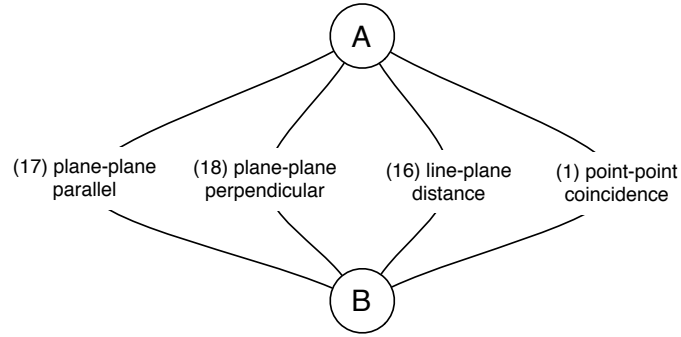


Figure 3: The cad graph for the example depicted in Figure 2.

6. point-plane distance
7. line-line parallel
8. line-line perpendicular
9. line-line fixed angular
10. line-line coincidence
11. line-line distance
12. line-plane parallel
13. line-plane perpendicular
14. line-plane fixed angular
15. line-plane coincidence

16. line-plane distance
17. plane-plane parallel
18. plane-plane perpendicular
19. plane-plane fixed angular
20. plane-plane coincidence
21. plane-plane distance

The geometric meaning of these colors will be described in the next section; the cad graph only captures the type of constraint imposed.

2.2.2 Body-and-cad frameworks

A *body-and-cad framework* $(G, c, L_1, \dots, L_{21})$ is a cad graph (G, c) along with a family of functions L_1, \dots, L_{21} describing the geometry of the structure, where the function L_i captures the constraints corresponding to edges with the i th color. Let $E_i = \{e \in E \mid c(e) = c_i\}$ be the set of c_i -colored edges.

For example, the function for plane-plane fixed angular constraints $L_{19} : E_{19} \rightarrow (\mathbb{R}^3 \times \mathbb{R}^3) \times (\mathbb{R}^3 \times \mathbb{R}^3) \times \mathbb{R}$ maps an edge $e = ij$ to a triple $((\mathbf{p}_i, \mathbf{d}_i), (\mathbf{p}_j, \mathbf{d}_j), \alpha)$ so that the planes $(\mathbf{p}_i, \mathbf{d}_i)$ and $(\mathbf{p}_j, \mathbf{d}_j)$ affixed to bodies i , respectively j , are constrained to have the angle α between them. The function for point-point coincidence constraints $L_1 : E_1 \rightarrow \mathbb{R}^3 \times \mathbb{R}^3$ maps an edge $e = ij$ to a pair of points $(\mathbf{p}_i, \mathbf{p}_j)$ affixed to bodies i , respectively j , that are constrained to be coincident. We will define the complete family of functions when analyzing them individually in Section 4.

A *realization* $G(\mathcal{T})$ of a body-and-cad framework $(G, c, L_1, \dots, L_{21})$ assigns a tuple of frames $\mathcal{T} = (T_1, \dots, T_n)$ for each vertex, satisfying the specified constraints. In this paper, we are *not* concerned with realization questions. We will always assume that a body-and-cad framework is given by a concrete realization, from which the family of functions L_1, \dots, L_{21} is computed.

Body-and-cad rigidity. Intuitively, a body-and-cad framework is *rigid* if the only motions respecting the constraints are the *trivial* 3D motions (rotations and translations); otherwise, it is *flexible*. We omit the technical definition, as it falls outside the scope of this paper.

Body-and-cad minimal rigidity. For classical distance constraints, the concept of *minimal rigidity* is defined as follows: a structure is minimally rigid if the removal of any constraint results in a flexible structure. However, in our case, geometric constraints may correspond to more than one “primitive” constraint. Intuitively, a *primitive* constraint yields only one row in the rigidity matrix (formally defined in Section 3), while the body-and-cad constraints may yield several rows. In our setting, we define minimal rigidity as above, but referring to the removal of primitive constraints only: a rigid body-and-cad structure is *minimally rigid* if the removal of any primitive constraint results in a flexible structure.

We return to the example from Figure 2 to illustrate the subtleties of this concept. The structure depicted is *rigid*. We say the structure is *overconstrained* since it remains rigid even after the removal of constraint (iii). The resulting structure is now minimally rigid. As we will see in Section 3, constraints (i), (ii) and (iv) correspond to 6 primitive constraints. Thus, the removal of any primitive constraint results in a flexible structure.

Now consider stacking the dice with the following two constraints: (i) (**line-line coincidence**) A 's Line 26 is coincident to B 's Line 12 and (ii) (**line-line coincidence**) A 's Line 36 is coincident to B 's Line 13. This structure is still rigid. While it becomes flexible after the removal of either constraint (i) or (ii), it is *not* minimally rigid. As we will see in Section 3, a line-line coincidence constraint corresponds to 4 primitive constraints. Thus, this structure has 8 primitive constraints and is overconstrained. To give some intuition, note that a structure composed of 2 rigid bodies has 12 degrees of freedom. Of these, 6 are trivial, so we may fix body A to factor them out. Now consider constraint (i); the structure is left with 2 degrees of freedom, as B may slide along the line and rotate about it. This line-line coincidence constraint is “eliminating” 4 degrees of freedom, formalized by the 4 rows of the rigidity matrix developed in Section 3 for the line-line coincidence constraint.

Body-and-cad infinitesimal rigidity. Infinitesimal rigidity is the linearized version of rigidity and is the only type we study in this paper. Let $\mathbf{s} = (\mathbf{s}_1, \dots, \mathbf{s}_n) \in (\mathbb{R}^6)^n$ assign an instantaneous screw \mathbf{s}_i to each body i and let $\mathbf{s}^* = (\mathbf{s}_1^*, \dots, \mathbf{s}_n^*)$. The vector \mathbf{s} is an *infinitesimal motion* of a body-and-cad structure if it infinitesimally respects the constraints. This can be expressed with the help of the *rigidity matrix*, fully described in Section 3. An infinitesimal motion is a vector in the kernel of the rigidity matrix. The kernel always contains the *trivial infinitesimal motions*, defined as those \mathbf{s} with $\mathbf{s}_i = \mathbf{s}_j$ for all i and j .

A body-and-cad framework is *infinitesimally rigid* if the only infinitesimal motions are trivial; otherwise, it is *infinitesimally flexible*.

Remarks. To develop the rigidity theory for a new model, three steps must be accomplished.

1. *Algebraic theory.* Formulate the rigidity concept in algebraic terms, resulting in an algebraic variety.
2. *Infinitesimal theory.* Analyze the local behavior at some point on the algebraic variety. This reduces to the study of a *rigidity matrix*.
3. *Combinatorial rigidity.* Seek a combinatorial characterization of minimal rigidity in terms of properties of an underlying graph structure. This is usually derived from properties of the rigidity matrix at a *generic* point on the algebraic variety.

In this paper, we directly formulate the infinitesimal rigidity theory for body-and-cad structures and identify combinatorial properties for the generic case.

To summarize, a *cad graph* is an edge-colored multigraph that captures the body-and-cad combinatorics, and a *body-and-cad framework* captures the geometry of the structure. As we develop the analysis of these concepts, an additional combinatorial object called the *primitive cad graph* will be associated to the cad graph. This is a multigraph with red or black edges, which captures certain combinatorial properties of infinitesimal body-and-cad rigidity.

3 Foundations of infinitesimal theory

The example from the previous section exposes some of the subtleties encountered with body-and-cad constraints that are not found when considering classical distance constraints. We introduce two new concepts to simplify the analysis: *primitive angular* and *blind* constraints. We then define, as building blocks, 4 *basic* angular and blind constraints and develop their infinitesimal theory. All 21 body-and-cad constraints can be studied using these **building blocks**, leading to the body-and-cad rigidity matrix.

3.1 Primitive constraints

A *primitive* constraint is one that may restrict at most one degree of freedom. For example, a **point-point distance** (bar) constraint is a primitive constraint, while the **line-line coincidence** constraint from the example in the preceding section is not. We classify primitive constraints into two types: *angular* and *blind*; as the theory is developed, it will become more clear why these classifications are appropriate, as they correspond to constraints demonstrating different algebraic behaviors.

A rigid body in 3D has 6 degrees of freedom, 3 of which are rotational and 3 of which are translational. A *primitive angular* constraint may restrict only a rotational degree of freedom, whereas a *primitive blind* constraint may restrict either a rotational or a translational degree of freedom. For instance, a **line-line perpendicular** constraint is a primitive angular constraint as it may restrict at most one rotational degree of freedom. A **point-point distance** (bar) constraint is a primitive blind constraint as it may restrict at most one rotational or translational degree of freedom. We will associate a set of primitive angular and a set of primitive blind constraints with each body-and-cad constraint.

3.2 Rigidity matrix

The rigidity matrix R for a body-and-cad structure has 6 columns for each body i , corresponding to the components of the instantaneous screw \mathbf{s}_i , as was done for the original body-and-bar rigidity matrix². There is a row for each primitive constraint associated to the original body-and-cad structure. A primitive angular constraint results in a row containing zero entries in the first 3 columns

²The starred version \mathbf{s}_i^* (see Section 2.1) will be used to conveniently order the columns of the rigidity matrix.

for each body, while a primitive blind constraint may have non-zero entries in any of the 6 columns for each body. In the schematic below, gray cells indicate potentially non-zero entries, and red cells highlight the zero entries for angular constraints.

Since the trivial motions corresponding to the 3D rigid motions are necessarily in the kernel of R , the maximum rank of R is $6n - 6$. By definition, a structure is *infinitesimally rigid* if its rigidity matrix has rank exactly $6n - 6$.

\mathbf{s}_1^*			\mathbf{s}_i^*			\mathbf{s}_n^*		
\mathbf{v}_1	$-\boldsymbol{\omega}_1$	\cdots	\mathbf{v}_i	$-\boldsymbol{\omega}_i$	\cdots	\mathbf{v}_n	$-\boldsymbol{\omega}_n$	
Angular constraints	0	\vdots	0	\vdots	\vdots	0	\vdots	
	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	
	0	\vdots	0	\vdots	\vdots	0	\vdots	
	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	
	0	\vdots	0	\vdots	\vdots	0	\vdots	
Blind constraints	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	

3.3 Building blocks

We now define 4 very specific **basic** angular and blind constraints (2 of each) and develop the infinitesimal theory for them. Everything in Section 4 is derived from these basic **building blocks**. The material presented here is the most technical part of our paper.

Angular building blocks

All body-and-cad angular constraints can be reduced to the following basic constraints between pairs of lines:

- (i) *basic line-line non-parallel fixed angular*, and
- (ii) *basic line-line parallel*.

(i) *Basic line-line non-parallel fixed angular*. A line-line non-parallel angular constraint between bodies i and j is defined by identifying a pair of non-parallel lines, each rigidly affixed to one body, and fixing the angle between them. Let \mathbf{d}_i and \mathbf{d}_j be the directions of the lines affixed to bodies i and j , respectively. Then the constraint is infinitesimally maintained if the axis of the relative screw $\mathbf{s}_i - \mathbf{s}_j$ is in a direction lying in the plane determined by \mathbf{d}_i and \mathbf{d}_j , i.e.,

$$\langle (\boldsymbol{\omega}_i - \boldsymbol{\omega}_j), \mathbf{d}_i \times \mathbf{d}_j \rangle = 0$$

Since $-\boldsymbol{\omega}_i$ is composed of the last three coordinates of \mathbf{s}_i^* , this is equivalent to

$$\langle (\mathbf{s}_i^* - \mathbf{s}_j^*), ((0, 0, 0), \mathbf{d}_j \times \mathbf{d}_i) \rangle = 0 \quad (3.1)$$

This corresponds to one row in the rigidity matrix:

$$\begin{array}{ccccccc}
& & \mathbf{s}_i^* & & \mathbf{s}_j^* & & \\
& \cdots & \mathbf{v}_i & -\boldsymbol{\omega}_i & \cdots & \mathbf{v}_j & -\boldsymbol{\omega}_j & \cdots \\
\cdots \mathbf{0} \cdots & \mathbf{0} & \mathbf{d}_j \times \mathbf{d}_i & \cdots \mathbf{0} \cdots & \mathbf{0} & \mathbf{d}_i \times \mathbf{d}_j & \cdots \mathbf{0} \cdots
\end{array}$$

(ii) *Basic line-line parallel constraint.* A line-line parallel constraint between bodies i and j is defined by identifying a pair of parallel lines, each rigidly affixed to one body, and restricting them to remain parallel. Let $\mathbf{d} = (a, b, c)$ be the direction of the parallel lines. Then the constraint is infinitesimally maintained if the axis of the relative screw $\mathbf{s}_i - \mathbf{s}_j$ is in the same direction as \mathbf{d} , i.e., $(\boldsymbol{\omega}_i - \boldsymbol{\omega}_j) = \alpha \mathbf{d}$, for some scalar α . This can be expressed by the following two linear equations, where $\boldsymbol{\omega} = \boldsymbol{\omega}_i - \boldsymbol{\omega}_j = (\omega^x, \omega^y, \omega^z)$:

$$\begin{aligned}
\omega^x b - \omega^y a &= 0 \\
\omega^y c - \omega^z b &= 0
\end{aligned}$$

Since $-\boldsymbol{\omega}_i$ is composed of the last three coordinates of \mathbf{s}_i^* , these are equivalent to

$$\langle (\mathbf{s}_i^* - \mathbf{s}_j^*), (0, 0, 0, -b, a, 0) \rangle \quad (3.2)$$

$$\langle (\mathbf{s}_i^* - \mathbf{s}_j^*), (0, 0, 0, 0, -c, b) \rangle \quad (3.3)$$

and correspond to **two rows** in the rigidity matrix:

$$\begin{array}{ccccccc}
& & \mathbf{s}_i^* & & \mathbf{s}_j^* & & \\
& \cdots & \mathbf{v}_i & -\boldsymbol{\omega}_i & \cdots & \mathbf{v}_j & -\boldsymbol{\omega}_j & \cdots \\
\cdots \mathbf{0} \cdots & \mathbf{0} & (-b, a, 0) & \cdots \mathbf{0} \cdots & \mathbf{0} & (b, -a, 0) & \cdots \mathbf{0} \cdots \\
\cdots \mathbf{0} \cdots & \mathbf{0} & (0, -c, b) & \cdots \mathbf{0} \cdots & \mathbf{0} & (0, c, -b) & \cdots \mathbf{0} \cdots
\end{array}$$

Blind building blocks

Let \mathbf{p} be a point and \mathbf{p}' its instantaneous velocity resulting from the instantaneous screw $\mathbf{s} \in \mathbb{R}^6$. Let $\mathbf{c} \in \mathbb{R}^3$ be an arbitrary direction vector. We either constrain the velocity \mathbf{p}' to be orthogonal or parallel to \mathbf{c} . This yields the remaining basic constraints:

(iii) *basic blind orthogonality* (see Figure 4a), and

(iv) *basic blind parallel* (see Figure 4b).

Expressing both of them becomes straightforward using the following fact:

Fact 1. *Let $\mathbf{s} \in \mathbb{R}^6$ be an instantaneous screw, $\mathbf{p} \in \mathbb{R}^3$ a point and \mathbf{p}' the velocity of \mathbf{p} under the screw motion \mathbf{s} . Then*

$$\mathbf{s} \vee (\mathbf{p} : 1) = (\mathbf{p}', -\langle \mathbf{p}, \mathbf{p}' \rangle) \quad (3.4)$$

and, for any $\mathbf{q} \in \mathbb{R}^3$ and $q^w \in \mathbb{R}$,

$$\mathbf{s} \vee (\mathbf{p} : 1) \vee (\mathbf{q} : q^w) = \langle \mathbf{p}', \mathbf{q} \rangle - q^w \langle \mathbf{p}, \mathbf{p}' \rangle \quad (3.5)$$

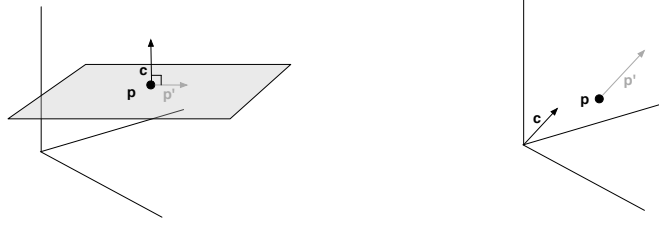
Proof. In the following, superscripts x, y, z, w denote the components of a vector in \mathbb{R}^4 . The minor of a 3×4 matrix A determined by columns i, j and k is denoted $|A_{ijk}|$.

If \mathbf{s} is a decomposable 2-tensor (a 2-extensor), then its components are the minors of a 2×4 matrix M ; see, e.g., [32, 33] for a standard review of 2-tensors in Grassmann-Cayley algebra. Let A be the 3×4 matrix obtained by appending $(\mathbf{p} : 1)$ to the bottom of M . The join $\mathbf{s} \vee (\mathbf{p} : 1)$ is the collection of the four minors of A . We fix the convention that $\mathbf{s} \vee (\mathbf{p} : 1) = (|A_{234}|, -|A_{134}|, |A_{124}|, -|A_{123}|)$. Then

$$\mathbf{s} \vee (\mathbf{p} : 1) \vee (\mathbf{q} : q^w) = \begin{vmatrix} & M & \\ p^x & p^y & p^z & 1 \\ q^x & q^y & q^z & q^w \end{vmatrix}$$

Performing a Laplace expansion along the 4th row of the matrix yields $q^x|A_{234}| - q^y|A_{134}| + q^z|A_{124}| - q^w|A_{123}| = q^x(\mathbf{s} \vee (\mathbf{p} : 1))^x + q^y(\mathbf{s} \vee (\mathbf{p} : 1))^y + q^z(\mathbf{s} \vee (\mathbf{p} : 1))^z + q^w(\mathbf{s} \vee (\mathbf{p} : 1))^w$.

Crapo and Whiteley [2] derived that $\mathbf{s} \vee (\mathbf{p} : 1) = (\mathbf{p}' : -\langle \mathbf{p}, \mathbf{p}' \rangle)$. Applying it, we obtain our desired result. The derivation when \mathbf{s} is indecomposable (the sum of two 2-extensors) is a simple extension obtained by working with the two 2-extensors simultaneously. \square



(a) *Orthogonality*: constrains the instantaneous velocity \mathbf{p}' of the point \mathbf{p} to be orthogonal to a direction \mathbf{c} . Then \mathbf{p}' must lie in the plane (\mathbf{p}, \mathbf{c}) .
(b) *Parallel*: constrains the instantaneous velocity \mathbf{p}' of a point \mathbf{p} to lie in the same direction as a vector \mathbf{c} .

Figure 4: Basic blind geometric constraints.

(iii) *Basic blind orthogonality constraint*

This constrains the instantaneous velocity \mathbf{p}' of the point \mathbf{p} to be orthogonal to a direction \mathbf{c} . To express this (see Figure 4a), we simply substitute $\mathbf{q} = \mathbf{c}$ and $q^w = 0$ into Equation 3.5. Then $\langle \mathbf{p}', \mathbf{c} \rangle = 0$ if and only if

$$\mathbf{s} \vee (\mathbf{p} : 1) \vee (\mathbf{c} : 0) = 0$$

if and only if

$$\langle \mathbf{s}^*, (\mathbf{p} : 1) \vee (\mathbf{c} : 0) \rangle = 0 \quad (3.6)$$

(iv) *Basic blind parallel constraint*

This constrains the instantaneous velocity \mathbf{p}' of the point \mathbf{p} to lie in the same direction as a direction \mathbf{c} . To express this (see Figure 4b), we apply Equation

3.5 twice by substituting $\mathbf{q} = (c^y, -c^x, 0)$ and $q^w = 0$ first, then $\mathbf{q} = (0, c^z, -c^y)$ and $q^w = 0$. We obtain that $\mathbf{p}' = \alpha \mathbf{c}$ for some $\alpha \in \mathbb{R}$ if and only if

$$\begin{aligned} \mathbf{s} \vee (\mathbf{p} : 1) \vee (c^y, -c^x, 0, 0) &= 0 \\ \mathbf{s} \vee (\mathbf{p} : 1) \vee (0, c^z, -c^y, 0) &= 0 \end{aligned}$$

if and only if

$$\langle \mathbf{s}^*, (\mathbf{p} : 1) \vee (c^y, -c^x, 0, 0) \rangle = 0 \quad (3.7)$$

$$\langle \mathbf{s}^*, (\mathbf{p} : 1) \vee (0, c^z, -c^y, 0) \rangle = 0 \quad (3.8)$$

4 Infinitesimal theory for body-and-cad constraints

We use the four basic building blocks just presented to complete the development of the infinitesimal theory. In this section, we present the rows of the rigidity matrix associated with each of the 21 body-and-cad constraints. In all figures, body i is represented by the green tetrahedron and body j by the purple cube.

4.1 Angular constraints

Angular constraints may be parallel, perpendicular or arbitrary fixed angular constraints; see Figures 5–7 for depictions of line-line, line-plane, and plane-plane angular constraints.

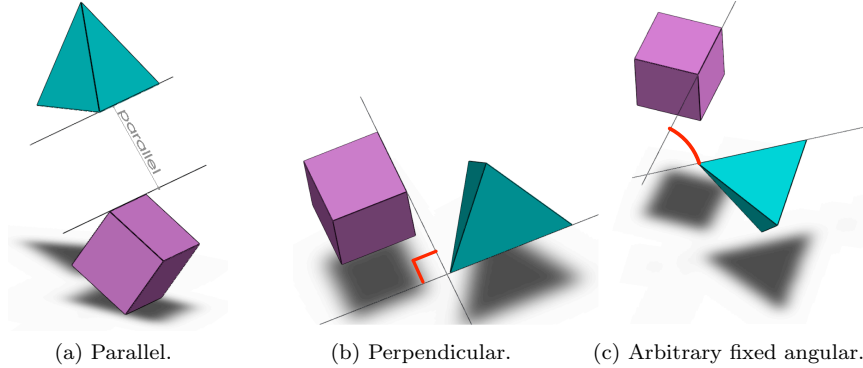


Figure 5: Line-line angular constraints.

We now formally define the functions for describing angular constraints.

- **Line-line parallel:** $L_7 : E_7 \rightarrow \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3$ maps an edge $e = ij$ to a triple $(\mathbf{p}_i, \mathbf{p}_j, \mathbf{d})$ so that the lines $(\mathbf{p}_i, \mathbf{d})$ and $(\mathbf{p}_j, \mathbf{d})$ affixed to bodies i and j , respectively, are constrained to remain parallel to each other.

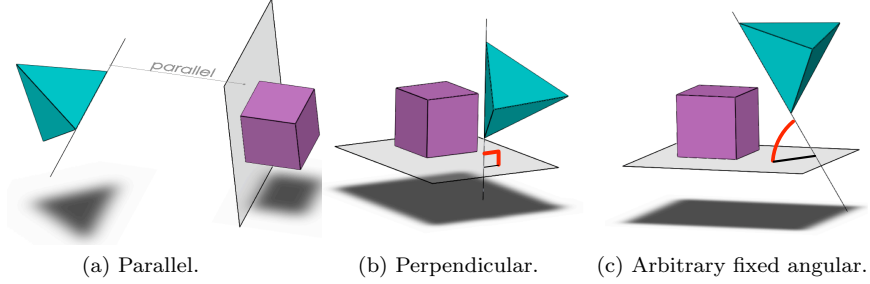


Figure 6: Line-plane angular constraints.

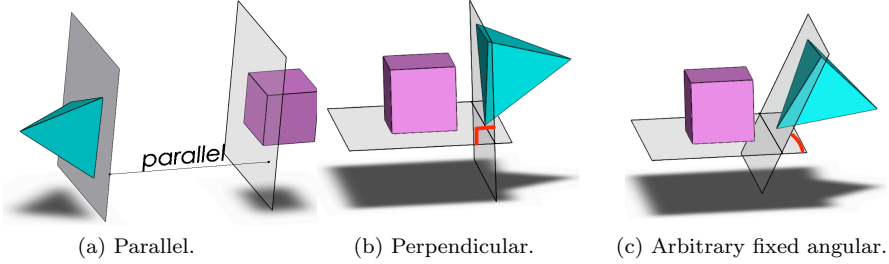


Figure 7: Plane-plane angular constraints.

- **Line-line perpendicular:** $L_8 : E_8 \rightarrow (\mathbb{R}^3 \times \mathbb{R}^3) \times (\mathbb{R}^3 \times \mathbb{R}^3)$ maps an edge $e = ij$ to a pair $((\mathbf{p}_i, \mathbf{d}_i), (\mathbf{p}_j, \mathbf{d}_j))$ so that the lines $(\mathbf{p}_i, \mathbf{d}_i)$ and $(\mathbf{p}_j, \mathbf{d}_j)$ affixed to bodies i and j , respectively, are constrained to remain perpendicular to each other.
- **Line-line fixed angular:** $L_9 : E_9 \rightarrow (\mathbb{R}^3 \times \mathbb{R}^3) \times (\mathbb{R}^3 \times \mathbb{R}^3) \times \mathbb{R}$ maps an edge $e = ij$ to a triple $((\mathbf{p}_i, \mathbf{d}_i), (\mathbf{p}_j, \mathbf{d}_j), \alpha)$ so that the lines $(\mathbf{p}_i, \mathbf{d}_i)$ and $(\mathbf{p}_j, \mathbf{d}_j)$ affixed to bodies i and j , respectively, are constrained to maintain the angle α between them.
- **Line-plane parallel:** $L_{12} : E_{12} \rightarrow (\mathbb{R}^3 \times \mathbb{R}^3) \times (\mathbb{R}^3 \times \mathbb{R}^3)$ maps an edge $e = ij$ to a pair $((\mathbf{p}_i, \mathbf{d}_i), (\mathbf{p}_j, \mathbf{d}_j))$ so that the line $(\mathbf{p}_i, \mathbf{d}_i)$ and plane $(\mathbf{p}_j, \mathbf{d}_j)$ affixed to bodies i and j , respectively, are constrained to remain parallel to each other.
- **Line-plane perpendicular:** $L_{13} : E_{13} \rightarrow (\mathbb{R}^3 \times \mathbb{R}^3) \times (\mathbb{R}^3 \times \mathbb{R}^3)$ maps an edge $e = ij$ to a pair $((\mathbf{p}_i, \mathbf{d}_i), (\mathbf{p}_j, \mathbf{d}_j))$ so that the line $(\mathbf{p}_i, \mathbf{d}_i)$ and plane $(\mathbf{p}_j, \mathbf{d}_j)$ affixed to bodies i and j , respectively, are constrained to remain perpendicular to each other.
- **Line-plane fixed angular:** $L_{14} : E_{14} \rightarrow (\mathbb{R}^3 \times \mathbb{R}^3) \times (\mathbb{R}^3 \times \mathbb{R}^3) \times \mathbb{R}$ maps an edge $e = ij$ to a triple $((\mathbf{p}_i, \mathbf{d}_i), (\mathbf{p}_j, \mathbf{d}_j), \alpha)$ so that the line $(\mathbf{p}_i, \mathbf{d}_i)$ and

plane $(\mathbf{p}_j, \mathbf{d}_j)$ affixed to bodies i and j , respectively, are constrained to maintain the angle α between them.

- **Plane-plane parallel:** $L_{17} : E_{17} \rightarrow \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3$ maps an edge $e = ij$ to a triple $(\mathbf{p}_i, \mathbf{p}_j, \mathbf{d})$ so that the planes $(\mathbf{p}_i, \mathbf{d})$ and $(\mathbf{p}_j, \mathbf{d})$ affixed to bodies i and j , respectively, are constrained to remain parallel to each other.
- **Plane-plane perpendicular:** $L_{18} : E_{18} \rightarrow (\mathbb{R}^3 \times \mathbb{R}^3) \times (\mathbb{R}^3 \times \mathbb{R}^3)$ maps an edge $e = ij$ to a pair $((\mathbf{p}_i, \mathbf{d}_i), (\mathbf{p}_j, \mathbf{d}_j))$ so that the planes $(\mathbf{p}_i, \mathbf{d}_i)$ and $(\mathbf{p}_j, \mathbf{d}_j)$ affixed to bodies i and j , respectively, are constrained to remain perpendicular to each other.
- **Plane-plane fixed angular:** $L_{19} : E_{19} \rightarrow (\mathbb{R}^3 \times \mathbb{R}^3) \times (\mathbb{R}^3 \times \mathbb{R}^3) \times \mathbb{R}$ maps an edge $e = ij$ to a triple $((\mathbf{p}_i, \mathbf{d}_i), (\mathbf{p}_j, \mathbf{d}_j), \alpha)$ so that the planes $(\mathbf{p}_i, \mathbf{d}_i)$ and $(\mathbf{p}_j, \mathbf{d}_j)$ affixed to bodies i and j , respectively, are constrained to maintain the angle α between them.

It is straightforward that the line-line angular constraints (perpendicular, fixed angular and parallel) are expressed by the two basic angular building blocks. We observe that the line-plane and plane-plane angular constraints reduce to them as follows.

- **Line-plane parallel:** Reduces to **line-line non-parallel fixed angular** using the normal to the plane.
- **Line-plane perpendicular:** Reduces to **line-line parallel** using the normal to the plane.
- **Line-plane fixed angular:** Reduces to **line-line non-parallel fixed angular** using the normal to the plane.
- **Plane-plane parallel:** Reduces to **line-line parallel** using the planes' normal.
- **Plane-plane perpendicular:** Reduces to **line-line non-parallel fixed angular** using the planes' normals.
- **Plane-plane fixed angular:** Reduces to **line-line non-parallel fixed angular** using the planes' normals.

4.2 Blind constraints

The remaining coincidence and distance constraints reduce to some combination of basic angular and basic blind constraints from Section 3.3. We consider them in the following order: **point-point**, **point-line**, **point-plane**, **line-line**, **line-plane** and **plane-plane**. Since a **point-point distance** constraint (*Figure 8b*) is a bar (see [30, 34]), we consider only the **point-point coincidence** constraint.

The function $L_1 : E_1 \rightarrow \mathbb{R}^3$ maps an edge $e = ij$ to a point \mathbf{p} so that it is constrained to lie on bodies i and j simultaneously.

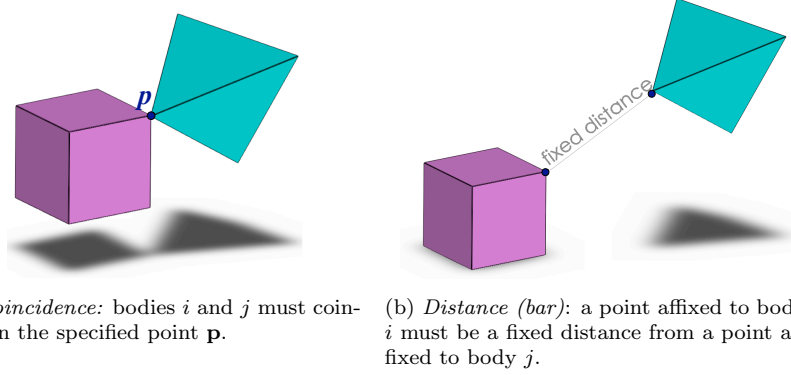


Figure 8: Point-point constraints.

Then the **point-point coincidence** constraint (*Figure 8a*) is infinitesimally maintained if the relative velocity of \mathbf{p} is exactly 0. Since $\mathbf{s}_i = (-\boldsymbol{\omega}_i, \mathbf{v}_i)$ and $\mathbf{s}_j = (-\boldsymbol{\omega}_j, \mathbf{v}_j)$, then the relative screw is defined by $(-\boldsymbol{\omega}, \mathbf{v})$, where $\boldsymbol{\omega} = (\boldsymbol{\omega}_i - \boldsymbol{\omega}_j)$ and $\mathbf{v} = (\mathbf{v}_i - \mathbf{v}_j)$. Therefore, the constraint is infinitesimally maintained if and only if \mathbf{p} 's infinitesimal velocity $\mathbf{p}' = \boldsymbol{\omega} \times \mathbf{p} + \mathbf{v} = 0$, i.e.,

$$\begin{aligned}\omega^y p^z - \omega^z p^y + v^x &= 0 \\ \omega^z p^x - \omega^x p^z + v^y &= 0 \\ \omega^x p^y - \omega^y p^x + v^z &= 0\end{aligned}$$

if and only if

$$\langle (s_i^* - s_j^*), (1, 0, 0, 0, -p^z, p^y) \rangle = 0 \quad (4.1)$$

$$\langle (s_i^* - s_j^*), (0, 1, 0, p^z, 0, -p^x) \rangle = 0 \quad (4.2)$$

$$\langle (s_i^* - s_j^*), (0, 0, 1, -p^y, p^x, 0) \rangle = 0 \quad (4.3)$$

Thus, a point-point coincidence constraint corresponds to 3 rows in the rigidity matrix:

			\mathbf{s}_i^*					\mathbf{s}_j^*				
			\mathbf{v}_i	$-\boldsymbol{\omega}_i$				\mathbf{v}_j	$-\boldsymbol{\omega}_j$			
\cdots	$\mathbf{0}$	\cdots	$(1, 0, 0, 0, -p^z, p^y)$		\cdots	$\mathbf{0}$	\cdots	$(-1, 0, 0, 0, p^z, -p^y)$		\cdots	$\mathbf{0}$	\cdots
\cdots	$\mathbf{0}$	\cdots	$(0, 1, 0, p^z, 0, -p^x)$		\cdots	$\mathbf{0}$	\cdots	$(0, -1, 0, -p^z, 0, p^x)$		\cdots	$\mathbf{0}$	\cdots
\cdots	$\mathbf{0}$	\cdots	$(0, 0, 1, -p^y, p^x, 0)$		\cdots	$\mathbf{0}$	\cdots	$(0, 0, -1, p^y, -p^x, 0)$		\cdots	$\mathbf{0}$	\cdots

Point-line coincidence (*Figure 9a*): The function $L_3 : E_3 \rightarrow \mathbb{R}^3 \times (\mathbb{R}^3 \times \mathbb{R}^3)$ maps an edge $e = ij$ to a pair $(\mathbf{p}_i, (\mathbf{p}_j, \mathbf{d}))$ so that point \mathbf{p}_i affixed to body i is constrained to lie on the line $(\mathbf{p}_j, \mathbf{d})$ affixed to body j .

The **point-line coincidence** constraint is infinitesimally maintained by using 2 primitive *blind* constraints from Equations 3.7 and 3.8 to express that the

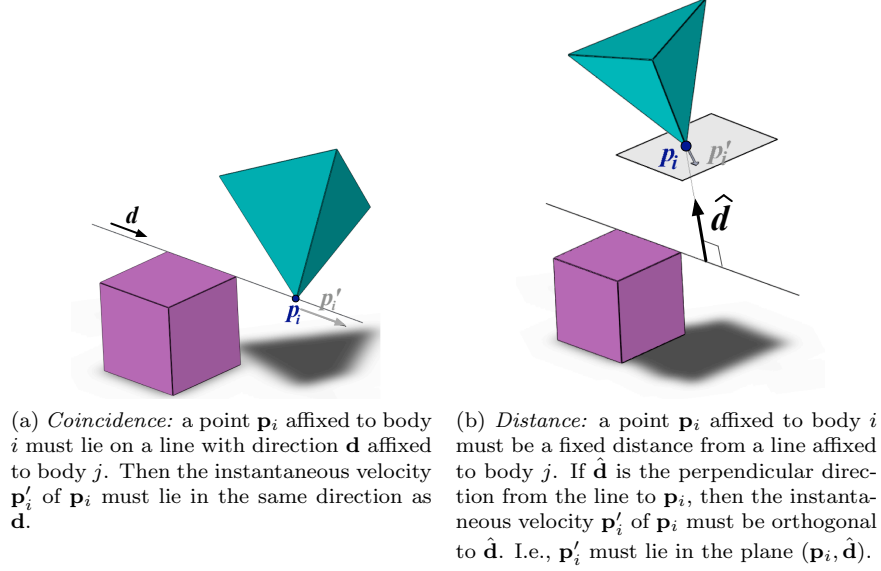


Figure 9: Point-line constraints.

relative velocity of \mathbf{p}_i lies in the same direction as \mathbf{d} :

$$\langle (\mathbf{s}_i - \mathbf{s}_j)^*, (\mathbf{p}_i : 1) \vee (d^y, -d^x, 0, 0) \rangle = 0 \quad (4.4)$$

$$\langle (\mathbf{s}_i - \mathbf{s}_j)^*, (\mathbf{p}_i : 1) \vee (0, d^z, -d^y, 0) \rangle = 0 \quad (4.5)$$

Thus, a point-line coincidence constraint corresponds to 2 rows in the rigidity matrix:

\mathbf{s}_i^*		\mathbf{s}_j^*	
\dots	$\mathbf{v}_i \quad -\boldsymbol{\omega}_i$	\dots	$\mathbf{v}_j \quad -\boldsymbol{\omega}_j$
$\dots \mathbf{0} \dots$	$(\mathbf{p}_i : 1) \vee (d^y, -d^x, 0, 0)$	$\dots \mathbf{0} \dots$	$-(\mathbf{p}_i : 1) \vee (d^y, -d^x, 0, 0)$
$\dots \mathbf{0} \dots$	$(\mathbf{p}_i : 1) \vee (0, d^z, -d^y, 0)$	$\dots \mathbf{0} \dots$	$-(\mathbf{p}_i : 1) \vee (0, d^z, -d^y, 0)$

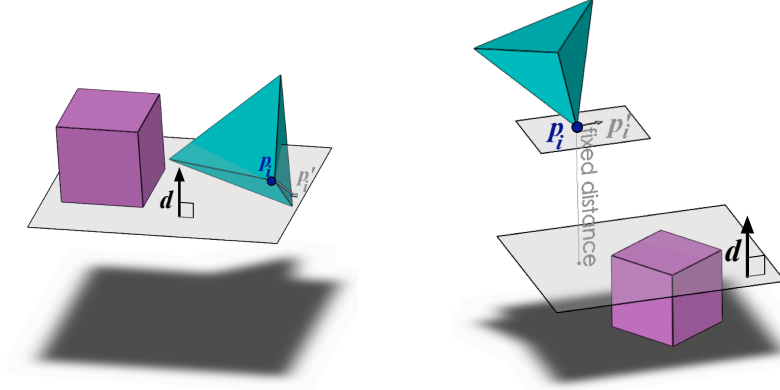
Point-line distance (*Figure 9b*): The function $L_4 : E_4 \rightarrow \mathbb{R}^3 \times (\mathbb{R}^3 \times \mathbb{R}^3) \times \mathbb{R}$ maps an edge $e = ij$ to a triple $(\mathbf{p}_i, (\mathbf{p}_j, \mathbf{d}), a)$ so that point \mathbf{p}_i affixed to body i is constrained to lie a distance a from the line $(\mathbf{p}_j, \mathbf{d})$ affixed to body j .

Let $\hat{\mathbf{d}}$ be the perpendicular direction from the line $(\mathbf{p}_j, \mathbf{d})$ to \mathbf{p}_i . Then the **point-line distance** constraint is infinitesimally maintained using 1 primitive *blind* constraint from Equation 3.6 to express that the relative velocity of \mathbf{p}_i is orthogonal to $\hat{\mathbf{d}}$:

$$\langle (\mathbf{s}_i - \mathbf{s}_j)^*, (\mathbf{p}_i : 1) \vee (\hat{\mathbf{d}} : 0) \rangle = 0 \quad (4.6)$$

Thus, a point-line distance constraint corresponds to one row in the rigidity matrix:

$$\begin{array}{ccccccc}
& & \overline{s_i^*} & & \overline{s_j^*} & & \\
\cdots & \mathbf{v}_i & -\boldsymbol{\omega}_i & \cdots & \mathbf{v}_j & -\boldsymbol{\omega}_j & \cdots \\
\cdots \mathbf{0} \cdots & (\mathbf{p}_i : 1) \vee (\hat{\mathbf{d}} : 0) & \cdots \mathbf{0} \cdots & -((\mathbf{p}_i : 1) \vee (\hat{\mathbf{d}} : 0)) & \cdots \mathbf{0} \cdots & &
\end{array}$$



(a) *Coincidence*: a point \mathbf{p}_i affixed to body i must lie in a plane with normal \mathbf{d} affixed to body j . Then the instantaneous velocity \mathbf{p}'_i of \mathbf{p}_i must remain in the plane $(\mathbf{p}_i, \mathbf{d})$.

(b) *Distance*: a point \mathbf{p}_i affixed to body i must be a fixed distance from a plane with normal \mathbf{d} affixed to body j . Then the instantaneous velocity \mathbf{p}'_i of \mathbf{p}_i must remain in the plane $(\mathbf{p}_i, \mathbf{d})$.

Figure 10: Point-plane constraints.

Point-plane coincidence (Figure 10a): The function $L_5 : E_5 \rightarrow \mathbb{R}^3 \times (\mathbb{R}^3 \times \mathbb{R}^3)$ maps an edge $e = ij$ to a pair $(\mathbf{p}_i, (\mathbf{p}_j, \mathbf{d}))$ so that the point \mathbf{p}_i affixed to body i is constrained to lie in the plane $(\mathbf{p}_j, \mathbf{d})$ affixed to body j .

The **point-plane coincidence** constraint is infinitesimally maintained using Equation 3.6 to express that the relative velocity of \mathbf{p}_i remains in the plane:

$$\langle (\mathbf{s}_i - \mathbf{s}_j)^*, (\mathbf{p}_i : 1) \vee (\mathbf{d} : 0) \rangle = 0 \quad (4.7)$$

Thus, a point-plane coincidence constraint corresponds to one row in the rigidity matrix:

$$\begin{array}{ccccccc}
& & \overline{s_i^*} & & \overline{s_j^*} & & \\
\cdots & \mathbf{v}_i & -\boldsymbol{\omega}_i & \cdots & \mathbf{v}_j & -\boldsymbol{\omega}_j & \cdots \\
\cdots \mathbf{0} \cdots & (\mathbf{p}_i : 1) \vee (\mathbf{d} : 0) & \cdots \mathbf{0} \cdots & -((\mathbf{p}_i : 1) \vee (\mathbf{d} : 0)) & \cdots \mathbf{0} \cdots & &
\end{array}$$

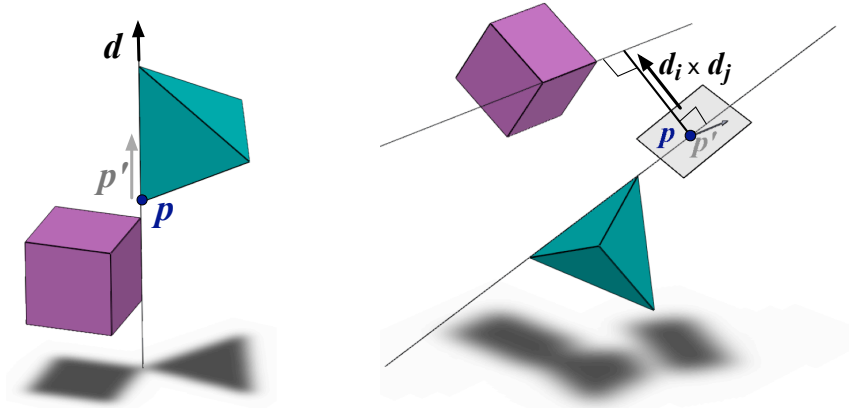
Point-plane distance (Figure 10b): The function $L_6 : E_6 \rightarrow \mathbb{R}^3 \times (\mathbb{R}^3 \times \mathbb{R}^3) \times \mathbb{R}$ maps an edge $e = ij$ to a triple $(\mathbf{p}_i, (\mathbf{p}_j, \mathbf{d}), a)$ so that the point \mathbf{p}_i affixed to body i is constrained to lie a distance a from the plane $(\mathbf{p}_j, \mathbf{d})$ affixed to body j .

The **point-plane distance** constraint is infinitesimally maintained by using Equation 3.6 to express that the relative velocity of \mathbf{p}_i remains parallel to the plane:

$$\langle (\mathbf{s}_i - \mathbf{s}_j)^*, (\mathbf{p}_i : 1) \vee (\mathbf{d} : 0) \rangle = 0 \quad (4.8)$$

Thus, a point-plane coincidence constraint corresponds to one row in the rigidity matrix:

			\mathbf{s}_i^*				\mathbf{s}_j^*			
\dots	\mathbf{v}_i	$-\boldsymbol{\omega}_i$		\dots	\mathbf{v}_j	$-\boldsymbol{\omega}_j$		\dots		
$\dots \mathbf{0} \dots$	$(\mathbf{p}_i : 1) \vee (\mathbf{d} : 0)$			$\dots \mathbf{0} \dots$	$-\left((\mathbf{p}_i : 1) \vee (\mathbf{d} : 0)\right)$			$\dots \mathbf{0} \dots$		



(a) *Coincidence*: bodies i and j must coincide on the specified line (\mathbf{p}, \mathbf{d}) . Then, in addition to a line-line parallel constraint, the instantaneous velocity \mathbf{p}' must lie in the same direction as \mathbf{d} .

(b) *Distance*: a line $(\mathbf{p}_i, \mathbf{d}_i)$ affixed to body i must be a fixed distance from a line $(\mathbf{p}_j, \mathbf{d}_j)$ affixed to body j . If \mathbf{p} is the point on the line $(\mathbf{p}_i, \mathbf{d}_i)$ that is closest to the line $(\mathbf{p}_j, \mathbf{d}_j)$, then the instantaneous velocity \mathbf{p}' of \mathbf{p} must be orthogonal to $\mathbf{d}_i \times \mathbf{d}_j$, the perpendicular to both lines. I.e., \mathbf{p}' must lie in the plane $(\mathbf{p}, \mathbf{d}_i \times \mathbf{d}_j)$.

Figure 11: Line-line constraints.

Line-line coincidence (Figure 11a): The function $L_{10} : E_{10} \rightarrow \mathbb{R}^3 \times \mathbb{R}^3$ maps an edge $e = ij$ to a pair (\mathbf{p}, \mathbf{d}) so that the line (\mathbf{p}, \mathbf{d}) is constrained to be affixed to bodies i and j simultaneously.

We place a **line-line parallel** angular constraint, resulting in 2 primitive *angular* constraints from Equations 3.2 and 3.3:

$$\langle (\mathbf{s}_i^* - \mathbf{s}_j^*), (0, 0, 0, -d^y, d^x, 0) \rangle \quad (4.9)$$

$$\langle (\mathbf{s}_i^* - \mathbf{s}_j^*), (0, 0, 0, 0, -d^z, d^y) \rangle \quad (4.10)$$

Then, to maintain coincidence, associate 2 primitive *blind* constraints from Equations 3.7 and 3.8 to force the relative velocity of \mathbf{p} to lie along \mathbf{d} :

$$\langle (\mathbf{s}_i - \mathbf{s}_j)^*, (\mathbf{p} : 1) \vee (d^y, -d^x, 0, 0) \rangle = 0 \quad (4.11)$$

$$\langle (\mathbf{s}_i - \mathbf{s}_j)^*, (\mathbf{p} : 1) \vee (0, d^z, -d^y, 0) \rangle = 0 \quad (4.12)$$

These 4 equations maintain the **line-line coincidence** constraint infinitesimally and correspond to 4 rows in the rigidity matrix:

\mathbf{s}_i^*			\mathbf{s}_j^*		
\dots	\mathbf{v}_i	$-\boldsymbol{\omega}_i$	\dots	\mathbf{v}_j	$-\boldsymbol{\omega}_j$
$\dots \mathbf{0} \dots$	$\mathbf{0}$	$(-d^y, d^x, 0)$	$\dots \mathbf{0} \dots$	$\mathbf{0}$	$(d^y, -d^x, 0)$
$\dots \mathbf{0} \dots$	$\mathbf{0}$	$(0, -d^z, d^y)$	$\dots \mathbf{0} \dots$	$\mathbf{0}$	$(0, d^z, -d^y)$
$\dots \mathbf{0} \dots$	$(\mathbf{p} : 1) \vee (d^y, -d^x, 0, 0)$		$\dots \mathbf{0} \dots$	$-((\mathbf{p} : 1) \vee (d^y, -d^x, 0, 0))$	
$\dots \mathbf{0} \dots$	$(\mathbf{p} : 1) \vee (0, d^z, -d^y, 0)$		$\dots \mathbf{0} \dots$	$-((\mathbf{p} : 1) \vee (0, d^z, -d^y, 0))$	

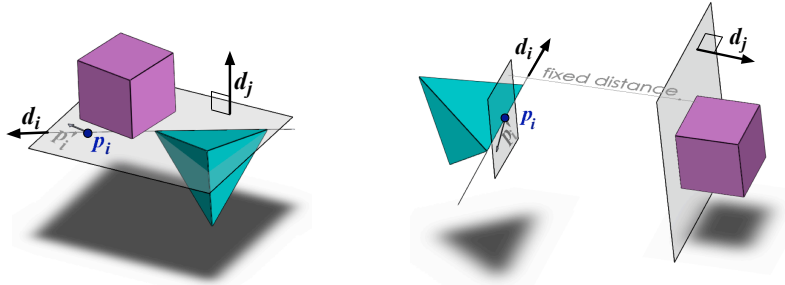
Line-line distance (Figure 11b): The function $L_{11} : E_{11} \rightarrow (\mathbb{R}^3 \times \mathbb{R}^3) \times (\mathbb{R}^3 \times \mathbb{R}^3) \times \mathbb{R}$ maps an edge $e = ij$ to a triple $((\mathbf{p}_i, \mathbf{d}_i), (\mathbf{p}_j, \mathbf{d}_j), a)$ so that the lines $(\mathbf{p}_i, \mathbf{d}_i)$ and $(\mathbf{p}_j, \mathbf{d}_j)$ affixed to bodies i and j , respectively, are constrained to lie a distance a from each other.

Let $\mathbf{p} \in \mathbb{R}^3$ be the point on the line $(\mathbf{p}_i, \mathbf{d}_i)$ closest to the line $(\mathbf{p}_j, \mathbf{d}_j)$. Then the **line-line distance** constraint is infinitesimally maintained if the relative velocity of \mathbf{p} is orthogonal to the direction perpendicular to both lines. In other words, \mathbf{p}' must lie in the plane defined by the point \mathbf{p} and normal direction $\mathbf{d}_i \times \mathbf{d}_j$. By substituting \mathbf{p} and $\mathbf{d}_i \times \mathbf{d}_j$ into Equation 3.6, we obtain the linear equation

$$\langle (\mathbf{s}_i - \mathbf{s}_j)^*, (\mathbf{p} : 1) \vee ((\mathbf{d}_i \times \mathbf{d}_j) : 0) \rangle = 0 \quad (4.13)$$

associating one *blind* primitive constraint. This corresponds to one row in the rigidity matrix:

\mathbf{s}_i^*			\mathbf{s}_j^*		
\dots	\mathbf{v}_i	$-\boldsymbol{\omega}_i$	\dots	\mathbf{v}_j	$-\boldsymbol{\omega}_j$
$\dots \mathbf{0} \dots$	$(\mathbf{p} : 1) \vee ((\mathbf{d}_i \times \mathbf{d}_j) : 0)$		$\dots \mathbf{0} \dots$	$-((\mathbf{p} : 1) \vee ((\mathbf{d}_i \times \mathbf{d}_j) : 0))$	



(a) *Coincidence*: a line $(\mathbf{p}_i, \mathbf{d}_i)$ affixed to body i must lie in a plane with normal \mathbf{d}_j affixed to body j . Then, in addition to a line-plane parallel constraint, the instantaneous velocity \mathbf{p}'_i of \mathbf{p}_i must lie in the plane $(\mathbf{p}_i, \mathbf{d}_j)$.

(b) *Distance*: a line $(\mathbf{p}_i, \mathbf{d}_i)$ affixed to body i must be a fixed distance from a plane with normal \mathbf{d}_j affixed to body j . Then, in addition to a line-plane parallel constraint, the instantaneous velocity \mathbf{p}'_i of \mathbf{p}_i must lie in the plane $(\mathbf{p}_i, \mathbf{d}_j)$.

Figure 12: Line-plane constraints.

Line-plane coincidence (*Figure 12a*): The function $L_{15} : E_{15} \rightarrow (\mathbb{R}^3 \times \mathbb{R}^3) \times (\mathbb{R}^3 \times \mathbb{R}^3)$ maps an edge $e = ij$ to a pair $((\mathbf{p}_i, \mathbf{d}_i), (\mathbf{p}_j, \mathbf{d}_j))$ so that the line $(\mathbf{p}_i, \mathbf{d}_i)$ affixed to body i is constrained to lie in the plane $(\mathbf{p}_j, \mathbf{d}_j)$ affixed to body j .

The **line-plane coincidence** constraint is infinitesimally maintained using a primitive *angular line-plane parallel* constraint from Equation 3.1:

$$\langle (\mathbf{s}_i^* - \mathbf{s}_j^*), ((0, 0, 0), \mathbf{d}_j \times \mathbf{d}_i) \rangle = 0 \quad (4.14)$$

In addition, a primitive *blind* constraint from Equation 3.6 forces the relative velocity of \mathbf{p}_i to remain in the plane:

$$\langle (\mathbf{s}_i - \mathbf{s}_j)^*, (\mathbf{p}_i : 1) \vee (\mathbf{d}_j : 0) \rangle = 0 \quad (4.15)$$

Thus, a line-plane coincidence constraint corresponds to 2 rows in the rigidity matrix:

\mathbf{s}_i^*			\mathbf{s}_j^*		
\dots	\mathbf{v}_i	$-\boldsymbol{\omega}_i$	\dots	\mathbf{v}_j	$-\boldsymbol{\omega}_j$
$\dots \mathbf{0} \dots$	0	$\mathbf{d}_j \times \mathbf{d}_i$	$\dots \mathbf{0} \dots$	0	$\mathbf{d}_i \times \mathbf{d}_j$
$\dots \mathbf{0} \dots$	$(\mathbf{p}_i : 1) \vee (\mathbf{d}_j : 0)$		$\dots \mathbf{0} \dots$	$-(\mathbf{p}_i : 1) \vee (\mathbf{d}_j : 0)$	

Line-plane distance (*Figure 12b*): The function $L_{16} : E_{16} \rightarrow (\mathbb{R}^3 \times \mathbb{R}^3) \times (\mathbb{R}^3 \times \mathbb{R}^3) \times \mathbb{R}$ maps an edge $e = ij$ to a triple $((\mathbf{p}_i, \mathbf{d}_i), (\mathbf{p}_j, \mathbf{d}_j), a)$ so that the line $(\mathbf{p}_i, \mathbf{d}_i)$ affixed to body i is constrained to lie a distance a from the plane $(\mathbf{p}_j, \mathbf{d}_j)$ affixed to body j .

The **line-plane distance** constraint is maintained infinitesimally by using the same equations as for the **line-plane coincidence** constraint: a primitive *angular line-plane parallel* constraint from Equation 3.1:

$$\langle (\mathbf{s}_i^* - \mathbf{s}_j^*), ((0, 0, 0), \mathbf{d}_j \times \mathbf{d}_i) \rangle = 0 \quad (4.16)$$

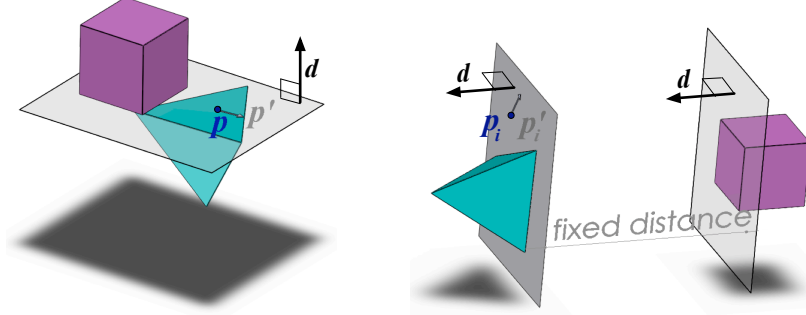
In addition, a primitive *blind* constraint from Equation 3.6 forces the relative velocity of \mathbf{p}_i to remain parallel to the plane:

$$\langle (\mathbf{s}_i - \mathbf{s}_j)^*, (\mathbf{p}_i : 1) \vee (\mathbf{d}_j : 0) \rangle = 0 \quad (4.17)$$

Thus, a line-plane distance constraint corresponds to 2 rows in the rigidity matrix:

\mathbf{s}_i^*			\mathbf{s}_j^*		
\dots	\mathbf{v}_i	$-\boldsymbol{\omega}_i$	\dots	\mathbf{v}_j	$-\boldsymbol{\omega}_j$
$\dots \mathbf{0} \dots$	0	$\mathbf{d}_j \times \mathbf{d}_i$	$\dots \mathbf{0} \dots$	0	$\mathbf{d}_i \times \mathbf{d}_j$
$\dots \mathbf{0} \dots$	$(\mathbf{p}_i : 1) \vee (\mathbf{d}_j : 0)$		$\dots \mathbf{0} \dots$	$-(\mathbf{p}_i : 1) \vee (\mathbf{d}_j : 0)$	

Plane-plane coincidence (*Figure 13a*): The function $L_{20} : E_{20} \rightarrow \mathbb{R}^3 \times \mathbb{R}^3$ maps an edge $e = ij$ to the pair (\mathbf{p}, \mathbf{d}) so that the plane (\mathbf{p}, \mathbf{d}) is constrained to be affixed to both bodies i and j simultaneously.



(a) *Coincidence*: bodies i and j must coincide on the specified plane (\mathbf{p}, \mathbf{d}) . Then, in addition to a plane-plane parallel constraint, the instantaneous velocity \mathbf{p}' of \mathbf{p} must remain in the plane.

(b) *Distance*: a plane $(\mathbf{p}_i, \mathbf{d})$ affixed to body i must be a fixed distance from a plane $(\mathbf{p}_j, \mathbf{d})$ affixed to body j . Then, in addition to a plane-plane parallel constraint, the instantaneous velocity \mathbf{p}'_i of \mathbf{p}_i must remain in the plane $(\mathbf{p}_i, \mathbf{d})$.

Figure 13: Plane-plane constraints.

We place a **plane-plane parallel** angular constraint, resulting in 2 primitive *angular* constraints from Equations 3.2 and 3.3:

$$\langle (\mathbf{s}_i^* - \mathbf{s}_j^*), (0, 0, 0, -d^y, d^x, 0) \rangle \quad (4.18)$$

$$\langle (\mathbf{s}_i^* - \mathbf{s}_j^*), (0, 0, 0, 0, -d^z, d^y) \rangle \quad (4.19)$$

Then, to maintain coincidence, place a primitive *blind* constraint by using Equation 3.6 to force the relative velocity of \mathbf{p} to remain in the plane:

$$\langle (\mathbf{s}_i - \mathbf{s}_j)^*, (\mathbf{p} : 1) \vee (\mathbf{d} : 0) \rangle = 0 \quad (4.20)$$

Thus, a plane-plane coincidence corresponds to 3 rows in the rigidity matrix:

\mathbf{s}_i^*			\mathbf{s}_j^*		
\dots	\mathbf{v}_i	$-\boldsymbol{\omega}_i$	\dots	\mathbf{v}_j	$-\boldsymbol{\omega}_j$
$\dots 0 \dots$	0	$(-d^y, d^x, 0)$	$\dots 0 \dots$	0	$(d^y, -d^x, 0)$
$\dots 0 \dots$	0	$(0, -d^z, d^y)$	$\dots 0 \dots$	0	$(0, d^z, -d^y)$
$\dots 0 \dots$	$(\mathbf{p} : 1) \vee (\mathbf{d} : 0)$		$\dots 0 \dots$	$-((\mathbf{p} : 1) \vee (\mathbf{d} : 0))$	

Plane-plane distance (Figure 13b): The function $L_{21} : E_{21} \rightarrow \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}$ maps an edge $e = ij$ to a quadruple $(\mathbf{p}_i, \mathbf{p}_j, \mathbf{d}, a)$ so that the planes $(\mathbf{p}_i, \mathbf{d})$ and $(\mathbf{p}_j, \mathbf{d})$ affixed to bodies i and j , respectively, are constrained to have the distance a between them.

Maintaining the plane-plane distance infinitesimally reduces to the same linear equations as for plane-plane coincidence. We place a **plane-plane parallel** angular constraint, resulting in 2 primitive *angular* constraints from Equations 3.2 and 3.3 along with a primitive *blind* constraint using Equation 3.6 to force

the relative velocity of \mathbf{p}_i to remain parallel to the plane:

$$\langle (\mathbf{s}_i^* - \mathbf{s}_j^*), (0, 0, 0, -d^y, d^x, 0) \rangle \quad (4.21)$$

$$\langle (\mathbf{s}_i^* - \mathbf{s}_j^*), (0, 0, 0, 0, -d^z, d^y) \rangle \quad (4.22)$$

$$\langle (\mathbf{s}_i - \mathbf{s}_j)^*, (\mathbf{p}_i : 1) \vee (\mathbf{d} : 0) \rangle = 0 \quad (4.23)$$

Thus, a plane-plane distance constraint corresponds to 3 rows in the rigidity matrix:

			\mathbf{s}_i^*			\mathbf{s}_j^*					
\dots			\mathbf{v}_i	$-\boldsymbol{\omega}_i$		\dots	\mathbf{v}_j	$-\boldsymbol{\omega}_j$		\dots	
$\dots \mathbf{0} \dots$	$\mathbf{0}$		$(-d^y, d^x, 0)$			$\dots \mathbf{0} \dots$	$\mathbf{0}$		$(d^y, -d^x, 0)$		$\dots \mathbf{0} \dots$
$\dots \mathbf{0} \dots$	$\mathbf{0}$		$(0, -d^z, d^y)$			$\dots \mathbf{0} \dots$	$\mathbf{0}$		$(0, d^z, -d^y)$		$\dots \mathbf{0} \dots$
$\dots \mathbf{0} \dots$			$(\mathbf{p}_i : 1) \vee (\mathbf{d} : 0)$			$\dots \mathbf{0} \dots$			$-(\mathbf{p}_i : 1) \vee (\mathbf{d} : 0)$		$\dots \mathbf{0} \dots$

4.2.1 Example

To help the reader, we complete the formalization of the dice example depicted in Figures 2 and 3. We assume that the z -axis lies along the base of the dice in the direction of Face 2, with the xy -plane parallel to Face 3.

Then the framework is described by the functions:

- $L_{17}(e_{(i)}) = ((0, 2, 0), (0, 1, 0), (0, 1, 0))$
- $L_{18}(e_{(ii)}) = (((0, 2, 0), (1, 0, 0)), ((0, 0, 1), (0, 0, 1)))$
- $L_{16}(e_{(iii)}) = (((0, 2, 0), (0, 0, 1)), ((0, 1, 0), (0, 1, 0)), 1)$
- $L_1(e_{(iv)}) = (0, 1, 1)$

Since the example only uses four types of constraints, we omit the description of the remaining L_i functions.

For each edge, we construct the associated rows in the rigidity matrix, resulting in the following:

\mathbf{s}_A^*						\mathbf{s}_B^*					
\mathbf{v}_A			$-\boldsymbol{\omega}_A$			\mathbf{v}_B			$-\boldsymbol{\omega}_B$		
$e_{(i)} \left\{ \right.$	0	0	0	-1	0	0	0	0	0	1	0
	0	0	0	0	0	1	0	0	0	0	-1
$e_{(ii)} \left\{ \right.$	0	0	0	0	1	0	0	0	0	0	-1
	0	0	0	1	0	0	0	0	0	-1	0
$e_{(iii)} \left\{ \right.$	0	-1	0	0	0	0	1	0	0	0	0
	1	0	0	0	-1	1	-1	0	0	0	-1
$e_{(iv)} \left\{ \right.$	0	1	0	1	0	0	0	-1	0	-1	0
	0	0	1	-1	0	0	0	-1	1	0	0

4.3 Summary of infinitesimal theory

We have now completed the development of the infinitesimal theory for body-and-cad rigidity. Table 1 summarizes the associations for each constraint to the number of primitive **angular** and **blind** constraints. As an example of how to read the table, the last two columns (corresponding to **plane**) of row 3 (corresponding to **coincidence** under **line**) indicate that a **line-plane coincidence** constraint reduces to 1 angular and 1 blind primitive constraint. In the next section, we identify a combinatorial property based on the shape of the rigidity matrix.

	point		line		plane	
	angular	blind	angular	blind	angular	blind
point						
coincidence	0	3	0	2	0	1
distance	0	1	0	1	0	1
line						
coincidence			2	2	1	1
distance			0	1	1	1
parallel			2	0	1	0
perpendicular			1	0	2	0
fixed angular			1	0	1	0
plane						
coincidence					2	1
distance					2	1
parallel					2	0
perpendicular					1	0
fixed angular					1	0

Table 1: Association of body-and-cad (*coincidence*, *angular*, *distance*) constraints with the number of blind and angular primitive constraints.

5 Combinatorics

Now we address the question of combinatorially characterizing when a body-and-cad rigidity matrix is generically independent, i.e., the rank function drops only on a measure-zero set of possible entries. The shape of the rigidity matrix leads to a natural property that we call *nested sparsity*. We show that nested sparsity is a necessary condition for body-and-cad rigidity and prove by a counterexample that it is insufficient.

Nested sparsity. A graph on n vertices is (k, ℓ) -sparse if every subset of n' vertices spans at most $kn' - \ell$ edges; it is *tight* if, in addition, it spans $kn - \ell$ total edges.

Let $G = (V, R \cup B)$ be a graph with its edge set colored into red and black edges, corresponding to R and B , respectively. We say that G is $(k_1, \ell_1, k_2, \ell_2)$ -*nested sparse* if it is (k_1, ℓ_1) -sparse and $G_R = (V, R)$ is (k_2, ℓ_2) -sparse; the graph is $(k_1, \ell_1, k_2, \ell_2)$ -*nested tight* if, in addition, G is (k_1, ℓ_1) -tight. Note that nested sparsity only makes sense when (k_2, ℓ_2) -sparsity is more restrictive than (k_1, ℓ_1) -sparsity.

Primitive cad graphs. Given a cad graph (G, c) , we define the *primitive cad graph* $H = (V, R \cup B)$ to be the multigraph obtained by assigning vertices to bodies and constraints to disjoint edge sets R and B , corresponding respectively to primitive angular and blind constraints. For each edge e with type c_i , associate primitive angular constraints to edges in R and primitive blind constraints to edges in B as described in Table 1. Figure 14 depicts the primitive cad graph associated with the dice example from Figure 2, whose cad graph is depicted in Figure 3.

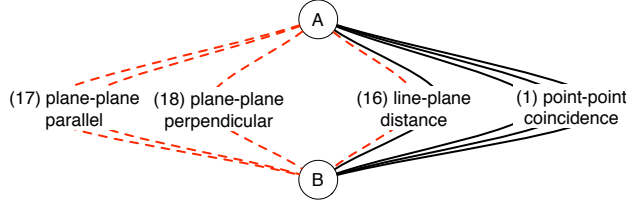
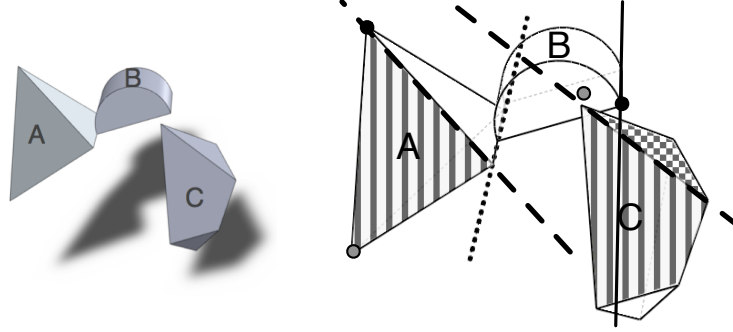


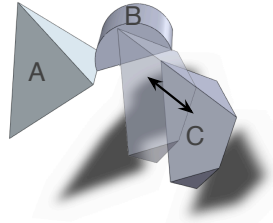
Figure 14: The primitive cad graph for the example depicted in Figures 2 and 3; dashed edges denote R edge set.

Theorem 5.1. *Let $H = (V, R \cup B)$ be the primitive cad graph associated to a body-and-cad framework, where R and B correspond to primitive angular and blind constraints, respectively. Then $(6, 6, 3, 3)$ -nested tightness is a necessary condition for generic minimal body-and-cad rigidity.*

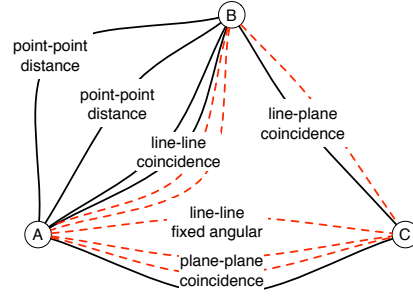
Proof. Let \mathbf{A} be the rigidity matrix associated with G . Reorder the columns so that the first $3n$ columns correspond to the $-\omega$ elements of the screws and the last $3n$ columns correspond to the \mathbf{v} elements. Reorder the rows to have the $|R|$ rows corresponding to primitive angular constraints first; since these rows have all 0s in the last $3n$ columns, we simply consider the submatrix \mathbf{A}_R defined by these $|R|$ rows and the first $3n$ columns. Then it is clear that $(3, 0)$ -sparsity is necessary on $G_R = (V, R)$. To see that $(3, 3)$ -sparsity is necessary, we note that the 3-dimensional space of trivial motions of $\mathfrak{so}(3)$ (infinitesimal rotations) is a subspace of the kernel of \mathbf{A}_R . These are defined by the basis $\{\boldsymbol{\rho}_1, \boldsymbol{\rho}_2, \boldsymbol{\rho}_3\}$, where $\boldsymbol{\rho}_1$ is the vector obtained by n copies of $(1, 0, 0)$, $\boldsymbol{\rho}_2$ is the vector obtained by n copies of $(0, 1, 0)$, and $\boldsymbol{\rho}_3$ is the vector obtained by n copies of $(0, 0, 1)$. Similarly, for the overall $(6, 6)$ -sparsity, note that we have a 6-dimensional space of trivial motions of $\mathfrak{se}(3)$ (infinitesimal rigid body motions), defined by the basis $\{\hat{\boldsymbol{\rho}}_1, \hat{\boldsymbol{\rho}}_2, \hat{\boldsymbol{\rho}}_3, \boldsymbol{\tau}_1, \boldsymbol{\tau}_2, \boldsymbol{\tau}_3\}$, where $\hat{\boldsymbol{\rho}}_i$ simply appends $3n$ zeros to $\boldsymbol{\rho}_i$; $\boldsymbol{\tau}_1$ is the vector



(a) Structure with 3 bodies and 6 constraints. A and B have 2 point-point distance constraints (denoted by the gray pair of points and black pair of points) and a line-line coincidence constraint (denoted by the shared dotted line). A and C have a line-line fixed angular constraint (denoted by the dashed lines) and a plane-plane coincidence constraint (denoted by the two faces with vertical stripes). B and C have a line-plane coincidence constraint (denoted by the solid line on B and the checkered face on C).



(b) The structure is flexible: C can move relative to A and B by translating in the direction indicated by the arrows.



(c) Corresponding primitive cad graph is $(6, 6, 3, 3)$ -nested tight; dashed edges denote R edge set.

Figure 15: Counterexample shows nested sparsity condition is not sufficient.

obtained by n copies of $(0, 0, 0, 1, 0, 0)$, τ_2 is the vector obtained by n copies of $(0, 0, 0, 0, 1, 0)$, and τ_3 is the vector obtained by n copies of $(0, 0, 0, 0, 0, 1)$. \square

Counterexample. We now show that $(6, 6, 3, 3)$ -nested sparsity is not sufficient for body-and-cad rigidity. The example in Figure 15 depicts a *flexible* structure whose associated graph is $(6, 6, 3, 3)$ -nested tight. It is composed of 3 bodies A, B and C ; Figure 15a depicts the constraints. The structure has one degree of freedom, as indicated by the arrows in Figure 15b. The associated primitive cad graph is shown in Figure 15c; the reader may check that it is $(6, 6, 3, 3)$ -nested tight.

6 Algorithms for nested sparsity

In the previous section, we defined nested sparsity, proving that is a necessary, but not sufficient, condition for body-and-cad rigidity. We now examine the algorithmic aspects of nested sparsity.

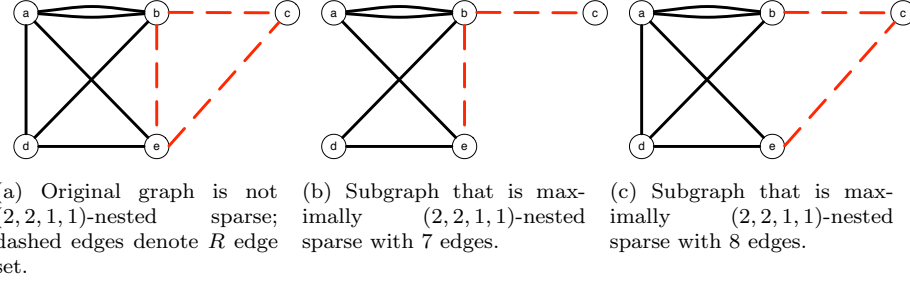


Figure 16: Example showing why $(2, 2, 1, 1)$ -sparsity is not matroidal; two maximal subgraphs do not have the same size.

We first observe that nested sparsity is not matroidal, as seen by the example for $(2, 2, 1, 1)$ -nested sparsity in Figure 16. However, for certain values of k_1, ℓ_1, k_2 and ℓ_2 , nested sparsity is the intersection of two matroids.

Theorem 6.1. [31] *When $0 \leq \ell_i < 2k_i$, for $i = 1, 2$, nested sparsity is the intersection of two matroids.*

Proof. Define the ground set E to be the complete graph $K_n^{(2k_1 - \ell_1) + (2k_2 - \ell_2)}$; the edges of the graph are colored red and black, with $2k_1 - \ell_1$ black edge multiplicity and $2k_2 - \ell_2$ red edge multiplicity. Let $M(k, \ell)$ be the bases of the (k, ℓ) -sparsity matroid; then $(k_1, \ell_1, k_2, \ell_2)$ -nested sparsity is the intersection of the following two matroids, defined by their bases:

1. $M_1 = \{E' \subseteq E \mid E' \in M(k_1, \ell_1)\}$, bases in the $M(k_1, \ell_1)$ -sparsity matroid when edge color is disregarded.
2. $M_2 = M(k_2, \ell_2) \cup K_n^{2k_1 - \ell_1}$, bases in the red (k_2, ℓ_2) -sparsity matroid padded with full edge multiplicity of the black edges.

□

As a consequence, when $0 \leq \ell_i < 2k_i$, the matroid intersection algorithm of Edmonds [4] can be used to solve the **Decision** (*is a graph nested sparse?*), **Extraction** (*given an input graph, extract a maximum-sized nested sparse subgraph*) and **Components** (*given an input graph, extract its maximal vertex sets that span nested tight subgraphs*) problems.

Edmonds' algorithm outputs a maximum-sized set of edges that are independent in both matroids and requires an *oracle* to test for independence in

each matroid. For the oracles, we use the pebble games algorithms of [14], a family of algorithms parametrized by two constants k and ℓ ; the (k, ℓ) -pebble game characterizes (k, ℓ) -sparsity. In particular, the (k, ℓ) -pebble game takes a graph as input and can be run in two modes: the **Decision** mode returns “yes” if the input graph is (k, ℓ) -sparse, and the **Components** mode returns the maximal vertex sets that span (k, ℓ) -tight subgraphs. Algorithm 1 gives a more detailed description of how Edmonds’ matroid intersection algorithm is used to solve problems for nested sparsity.

Algorithm 1. $(k_1, \ell_1, k_2, \ell_2)$ -nested sparsity: **Decision, Extraction and Components**

Input: A graph $G = (V, E = R \cup B)$ and constants k_1, ℓ_1, k_2, ℓ_2 , where $0 \leq \ell_i < 2k_i$ for $i = 1, 2$.

Method:

1. Run Edmonds’ matroid intersection algorithm [4] on G for the two matroids M_1 and M_2 , as defined in the proof of Theorem 6.1. When the algorithm performs independence queries on a set of edges $I \subseteq E$,
 - (a) For the matroid M_1 , play the (k_1, ℓ_1) -pebble game on the input (V, I) in **Decision** mode, which returns “yes” if it is (k_1, ℓ_1) -sparse
 - (b) For the matroid M_2 , play the (k_2, ℓ_2) -pebble game on the input $(V, I \cap R)$ in **Decision** mode, which returns “yes” if it is (k_2, ℓ_2) -sparse
2. Edmonds’ algorithm returns a set $I \subseteq E$ that is of maximum size, where (V, I) is $(k_1, \ell_1, k_2, \ell_2)$ -nested sparse.
3. Output:
 - For **Decision**, “yes” if $I = E$ and no otherwise.
 - For **Extraction**, (V, I) .
 - For **Components**, play the (k_1, ℓ_1) -pebble game in **Components** mode on (V, I) and output the components returned by the pebble game.

Figure 17: Algorithm for nested sparsity.

Complexity analysis. Edmonds’ algorithm queries the oracles $O(mr^2)$ times, where m is the number of elements in the ground set and r is the smaller rank of the two matroids. For nested sparsity, on a graph G with n vertices and $m = O(n^2)$ edges, both matroids have rank $O(n)$. Therefore, Edmonds’

algorithm requires $O(n^4)$ oracle queries. The pebble game algorithms require $O(n^2)$ time, resulting in $O(n^6)$ total complexity for Algorithm 1. We note that, using the recent matroid intersection algorithm of Harvey [8], a more efficient running time of $O(mr^{\omega-1}) = O(n^{\omega+1})$, where ω is the matrix multiplication exponent, can be obtained for nested sparsity.

Since $(6, 6, 3, 3)$ -nested sparsity meets the conditions for Theorem 6.1, we can apply Algorithm 1 to address the necessary condition for body-and-cad rigidity in polynomial time.

7 Conclusions

Constraint-based CAD software contains a rich set of geometric constraints. Motivated by their applications, we have initiated the study of body-and-cad rigidity by identifying a class of constraints amenable to rigidity-theoretical investigation and developing their infinitesimal theory. The shape of the rigidity matrix naturally led to the study of $(6, 6, 3, 3)$ -nested sparsity, a necessary, but not sufficient, condition for body-and-cad rigidity. The polynomial time algorithm we presented for testing this condition may have practical applications as a filter for finding rigid components in a CAD environment, providing informative feedback to the user.

Applications. The results presented can be applied to a larger set of CAD constraints via simple reductions. In particular, it is easy to establish the following reductions:

- **Sphere-sphere tangency:** Reduces to **point-point distance** using the sphere centers and the sum of the radii.
- **Sphere-plane tangency:** Reduces to **point-plane distance** using the sphere center, the plane and the sphere radius.
- **Sphere-line tangency:** Reduces to **point-line distance** using the sphere center, the line and the sphere radius.
- **Sphere-point coincidence:** Reduces to **point-point distance** using the sphere center, the point and the sphere radius.

Analogous reductions can be applied when considering cylinders instead of spheres by substituting the cylinder’s center axis for the sphere’s center point.

Future directions. It remains an open problem to find a combinatorial characterization for generic body-and-cad rigidity. We anticipate the study of some of the constraints introduced here may prove more tractable than classical 3D bar-and-joint rigidity. A full combinatorial characterization for angular constraints appears in [16, 13]. However, we observe that finding a complete characterization may require overcoming well-known obstacles such as detecting dependencies in 3D bar-and-joint, 2D points-and-angles, 2D circles-and-angles, and 2D point-line incidence constraint systems.

Analogous body-and-cad structures for 2D consist of rigid bodies with pairwise coincidence (point-point, point-line and line-line), angular (line-line) and distance (point-point, point-line and line-line) constraints identified between points and lines rigidly attached to bodies. The development of the rigidity matrix is a straightforward extension of this work. The interesting question, which remains future work, is a full combinatorial characterization.

Acknowledgements. The authors would like to thank Louis Theran for the observation and proof of Theorem 6.1. The figures were made using the SolidWorks CAD software [3].

References

- [1] S. R. Ball. *A Treatise on the Theory of Screws*. Cambridge University Press, Cambridge, 1900.
- [2] H. Crapo and W. Whiteley. The statics of frameworks and movements of panel structures, a geometric introduction. *Structural Topology*, 6:43–82, 1982.
- [3] Dassault Systèmes. Solidworks: 3d mechanical design and 3d CAD software, 2010. <http://www.solidworks.com/>.
- [4] J. Edmonds. Matroid intersection. *Annals of Discrete Mathematics*, 4:39–49, 1979.
- [5] X.-S. Gao, D. Lei, Q. Liao, and G.-F. Zhang. Generalized stewart platforms and their direct kinematics. *IEEE Trans. Robotics*, 21:141–151, 2005.
- [6] X.-S. Gao, Q. Lin, and G.-F. Zhang. A c-tree decomposition algorithm for 2d and 3d geometric constraint solving. *Computer-Aided Design*, 38(1):1–13, 2006.
- [7] J. Graver, B. Servatius, and H. Servatius. *Combinatorial rigidity*. Graduate Studies in Mathematics. American Mathematical Society, 1993.
- [8] N. J. A. Harvey. Algebraic algorithms for matching and matroid problems. *SIAM J. Comput.*, 39(2):679–702, 2009.
- [9] C. M. Hoffmann, A. Lomonosov, and M. Sitharam. Decomposition plans for geometric constraint problems, part II: New algorithms. *Journal of Symbolic Computation*, 31(4):409–427, 2001.
- [10] C. M. Hoffmann, A. Lomonosov, and M. Sitharam. Decomposition plans for geometric constraint systems, part I: performance measures for CAD. *Journal of Symbolic Computation*, 31(4):367–408, 2001.
- [11] C. Jermann, G. Trombettoni, B. Neveu, and P. Mathis. Decomposition of geometric constraint systems: a survey. *Int. J. Comput. Geometry Appl.*, 16(5-6):379–414, 2006.

- [12] G. Laman. On graphs and rigidity of plane skeletal structures. *Journal of Engineering Mathematics*, 4:331–340, 1970.
- [13] A. Lee. *Geometric Constraint Systems with Applications in CAD and Biology*. PhD thesis, University of Massachusetts Amherst, May 2008.
- [14] A. Lee and I. Streinu. Pebble game algorithms and sparse graphs. *Discrete Mathematics*, 308(8):1425–1437, April 2008.
- [15] A. Lee, I. Streinu, and L. Theran. Graded sparse graphs and matroids. *Journal of Universal Computer Science*, 13(11):1671–1679, 2007.
- [16] A. Lee-St. John and I. Streinu. Angular rigidity in 3d: combinatorial characterizations and algorithms. In *Proc. 21st Canadian Conf. Comp. Geometry (CCCG’09)*. Univ. British Columbia, August 2009.
- [17] H. Li and Y. Wu. Automated short proof generation for projective geometric theorems with Cayley and bracket algebras: I. incidence geometry. *J. Symb. Comput.*, 36(5):717–762, 2003.
- [18] J. C. Maxwell. On the calculation of the equilibrium and stiffness of frames. *Philos. Mag.*, 27:294, 1864.
- [19] D. Michelucci and P. Schreck. Incidence constraints: a combinatorial approach. *International Journal of Computational Geometry & Applications*, 16(5-6):443–460, 2006.
- [20] R. Plateaux, O. Penas, F. Mhenni, J.-Y. Choley, and A. Riviere. Introduction of the 3d geometrical constraints in Modelica. *Proceedings of the 7th International Modelica Conference, Como, Italy*, 2009.
- [21] F. Saliola and W. Whiteley. Constraining plane configurations in CAD: Circles, lines, and angles in the plane. *SIAM Journal on Discrete Mathematics*, 18(2):246–271, 2004.
- [22] J. M. Selig. *Geometrical Methods in Robotics*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1996.
- [23] P. Serre, M. Moinet, and A. Clement. Declaration and specification of a geometrical part in the language of geometric algebra. *Advanced Mathematical and Computational Tools in Metrology and Testing VIII*, 78:298–308, 2009.
- [24] B. Servatius and W. Whiteley. Constraining plane configurations in CAD: Combinatorics of directions and lengths. *SIAM Journal on Discrete Mathematics*, 12(1):136–153, 1999.
- [25] M. Sitharam. Combinatorial approaches to geometric constraint solving: Problems, progress, and directions. In R. Janardan, M. Smid, and D. Dutta, editors, *Geometric and algorithmic aspects of computer-aided design and manufacturing*, volume 67 of *DIMACS Ser. in Discrete Math. and Theoret. Comp. Sci.*, pages 117–163, Providence, RI, 2005. AMS.

- [26] M. Sitharam. Well-formed systems of point incidences for resolving collections of rigid bodies. *Int. J. Comput. Geometry Appl.*, 16(5-6):591–615, 2006.
- [27] M. Sitharam, J. Peters, and Y. Zhou. Optimized parametrization of systems of incidences between rigid bodies. *Journal of Symbolic Computation*, 45(4):481–498, April 2010.
- [28] M. Sitharam, Y. Zhou, and J. Peters. Reconciling conflicting combinatorial preprocessors for geometric constraint systems. *To appear in: International Journal of Computational Geometry and Applications*, 2010.
- [29] I. Streinu and L. Theran. Sparse hypergraphs and pebble game algorithms. *European Journal of Combinatorics*, 30(8):1944–1964, November 2009.
- [30] T.-S. Tay. Rigidity of multigraphs I: linking rigid bodies in n -space. *Journal of Combinatorial Theory Series, B* 26:95–112, 1984.
- [31] L. Theran. Personal communication, 2007.
- [32] N. White. Grassmann-Cayley algebra and robotics. *Journal of Intelligent and Robotics Systems*, 11:91–107, 1994.
- [33] N. White. Geometric applications of the Grassmann-Cayley algebra. In J. E. Goodman and J. O’Rourke, editors, *Handbook of Discrete and Computational Geometry*, CRC Press, 1997. 1997.
- [34] N. White and W. Whiteley. The algebraic geometry of motions of bar-and-body frameworks. *SIAM Journal of Algebraic Discrete Methods*, 8:1–32, 1987.
- [35] W. Whiteley. A matroid on hypergraphs, with applications in scene analysis and geometry. *Discrete and Computational Geometry*, 4:75–95, 1989.
- [36] W. Whiteley. Some matroids from discrete applied geometry. In J. Bonin, J. G. Oxley, and B. Servatius, editors, *Matroid Theory*, volume 197 of *Contemporary Mathematics*, pages 171–311. American Mathematical Society, 1996.
- [37] Y. Zhou. *Combinatorial Decomposition, Generic Independence and Algebraic Complexity of Geometric Constraints Systems: Applications in Biology and Engineering*. PhD thesis, University of Florida, 2006.